Title
A Fundamental Limitation of Symbol-Argument-Argument Notation As a Model of Human Relational Representations

Journal
Proceedings of the Annual Meeting of the Cognitive Science Society, 26(26)

ISSN
1069-7977

Authors
Doumas, Leonidas A.A.
Hummel, John E.

Publication Date
2004

Peer reviewed

A Fundamental Limitation of Symbol-Argument-Argument Notation As a Model of Human Relational Representations

Leonidas A. A. Doumas (adoumas@psych.ucla.edu)
John E. Hummel (jhummel@psych.ucla.edu)
Department of Psychology, University of California, Los Angeles
405 Hilgard Ave.
Los Angeles, CA 90095-1563

Abstract

Human mental representations are both structure-sensitive (i.e., symbolic) and semantically rich. Connectionist models have well-known limitations in capturing the structure-sensitivity of mental representations, while traditional symbolic models based on varieties of symbol-argument-argument notation (SAA) have difficulty capturing their semantic richness. We argue that this limitation of SAA is fundamental and cannot be alleviated in the notational format of SAA itself. Finally, we review an approach to human mental representation that captures both its structure-sensitivity and semantic richness.

Relational reasoning—reasoning constrained by the relational roles that objects play, rather than just the features of the objects themselves—is ubiquitous in human mental life, and includes analogical inference, schema induction, and the application of explicit rules (Gentner, 1983; Holyoak & Thagard, 1995). In order to support human-like relational thinking, a representational system must meet two general requirements (Hummel & Holyoak, 1997): First, it must represent relations independently of their fillers and simultaneously specify how fillers are bound to relational roles (i.e., it must be a *symbol system*; Newell, 1990). Second, it must explicitly specify the semantic content of relational roles and their fillers. In this paper, we consider in detail the implications of the latter requirement, with an emphasis on symbol-argument-argument notation (SAA), which includes propositional notation, high-rank tensors, and many varieties of labeled graphs.

Properties of Relational Representations

Human Relational Representations are Symbolic

Symbolic representations have the property that symbols are invariant with their role in an expression[1], and the meaning of the expression as a whole is a function of both the symbols and their arrangement (i.e., role-filler bindings). For example, the expressions *chase* (Pat, Don) and *chase* (Don, Pat) mean different things, but Pat, Don, and *chase* mean the same things in both expressions. Formal symbol systems have this property by assumption: It is given in the definition of the system that symbols retain their meanings across different expressions, and that the meaning of an expression is a function of both its constituent symbols and their arrangement. Physical symbol systems (Newell, 1990)—such as digital computers and human brains—cannot simply "assume" these properties, but instead must actively work to ensure that both are satisfied. The claim that human mental representations are symbolic in this sense is controversial (e.g., Elman et al., 1996), however, relational generalization has proved unattainable for non-symbolic models of cognition, and there is reason to believe it is fundamentally unattainable for such models (see Doumas & Hummel, in press; Halford, Wilson, & Phillips, 1998; Hummel & Holyoak, 2003a; Marcus, 1998).

Human Relational Representations Specify the Semantic Content of Objects and Relational Roles

A second important property of human relational representations is that they explicitly specify the semantic content of objects and relational roles (e.g., the *lover* and *beloved* roles of *love* (x, y) or the *killer* and *killed* roles of *murder* (x, y)): We know what it means to be a lover or a killer, and that knowledge is part of our representation of the relation itself (as opposed to being specified in a lookup table, a set of inference rules, or some other external structure). As a result, it is easy to appreciate that the patient (i.e., *killed*) role of *murder* (x, y) is like the patient role of *manslaughter* (x, y), even though the agent roles differ (i.e., the act is intentional in the former case but not the latter); and the agent (i.e., *killer*) role of *murder* (x, y) is similar to the agent role of *attempted-murder* (x, y), even though the patient roles differ.

More evidence that we represent the semantics of relational roles explicitly is that we can easily solve mappings that violate the "n-ary" restriction (Hummel & Holyoak, 1997). That is, we can map n-place predicates onto m-place predicates where $n \neq m$. For instance, given statements such as *taller-than* (Abe, Bill), *tall* (Chad) and *short* (Dave), it is easy to map Abe to Chad and Bill to

[1] This is not to say that symbols may not have shades of meaning that vary from one context to another. For example, the symbol "*loves*" suggests different relations in *loves* (Mary, John) vs. *loves* (Mary, Chocolate). However, as noted by Hummel and Holyoak (2003a), this kind of contextual shading must be a function of the system's knowledge, rather than an inevitable consequence of the way in which it binds relational roles to their fillers.

Dave. Given *shorter-than* (Eric, Fred), it is also easy to map Eric to Bill (and Dave) and Fred to Abe (and Chad). These mappings are based on the semantics of the individual relational roles rather than the formal syntax of propositional notation, or, say, the fact that *taller-than* and *shorter-than* are logical opposites: *love* (x, y) is in some sense the opposite of *hate* (x, y), but in contrast to *taller-than* and *shorter-than* (in which the first role of one relation maps to the second role of the other) the first role of *love* (x, y) more naturally maps to the first role of *hate* (x, y). In short, the similarity and/or mappings of various relational roles are idiosyncratic, based on the semantic content of the individual roles in question. The semantics of relational roles matter, and are an explicit part of the mental representation of relations.

The semantic properties of relational roles are also evidenced in numerous other ways in human cognition, including memory retrieval (e.g., Gentner, Ratterman & Forbus, 1993; Ross, 1989), and analogical mapping and inference (Bassok, Wu & Olseth, 1995; Kubose, Holyoak & Hummel, 2002; Krawczyk, Holyoak & Hummel, in press; Ross, 1987). Indeed, the meanings of relational roles influence relational thinking even when they are irrelevant or misleading (e.g., Bassok et al., 1995; Ross, 1989), suggesting that access to and use of role-based semantic information is quite automatic. This information appears to be an integral part of the mental representation of relations. Given its centrality in human cognition, an important criterion for a general account of human mental representation is that it must represent relations in a way that captures the semantics of their roles.

SAA Accounts of Relational Representations

Numerous models of human cognition account for the symbolic nature of relational representations by postulating representations based on varieties of symbol-argument-argument notation (SAA). These include propositional notation, varieties of labeled graphs, and high-rank tensor products (e.g., Anderson & Lebiere, 1998; Eliasmith & Thagard, 2001; Falkenhainer, Forbus, & Gentner, 1989; Forbus, Gentner, & Law, 1995; Halford et al., 1998; Holyoak & Thagard, 1989; Keane, Ledgeway, & Duff, 1994; Ramscar & Yartlett, 2003; Salvucci & Anderson, 2001). In SAA notation relations and their fillers are represented as independent symbolic units, which are bound together to form larger relational structures. For example, in propositional notation the statement "John loves Mary" is represented by binding the symbol for the predicate *love* to a list of its arguments, here John and Mary, forming the proposition *love* (John, Mary). By virtue of their positions in the list of arguments, John is taken to be the filler of the first role of the *loves* relation (i.e., the *lover*), and Mary is the filler of the second role (i.e., the *beloved*). Labeled graphs use location in the graph to indicate relational bindings in much the same way: Relations are represented independently of their arguments (fillers), and relation-filler bindings are represented in terms of the fillers' locations in the graph. As such, SAA is meaningfully symbolic in the sense described above.

Because SAA systems are meaningfully symbolic they naturally support operations that require relational representations, such as structure mapping (a.k.a., analogical mapping; see e.g., Falkenhainer et al., 1989; Forbus et al., 1995) and matching symbolic rules (see Anderson & Lebiere, 1998). This is no small accomplishment. Numerous representational schemes, including traditional distributed connectionist representations (e.g., Elman, et al., 1996) and the kinds of representations formed by latent semantic analysis (e.g., Landauer & Dumais, 1997) have not succeeded in modeling relational perception or cognition, and, as noted above, there are compelling reasons to believe they are fundamentally ill-suited for doing so.

The successes of SAA notation as a model of human mental representation are thus decidedly non-trivial. At the same time, however, SAA notation has greater difficulty capturing the semantic content of human mental representations. Indeed, this limitation was a central part of the criticisms leveled against symbolic modeling by the connectionists in the mid-1980s (e.g., Rumelhart, McClelland, & the PDP Research Group, 1986; for a more recent treatment, see Hummel & Holyoak, 1997). One of the strengths of distributed connectionist representations is their natural ability to capture the semantic content of the objects they represent. By contrast, propositional notation and labeled graphs have difficulty making this semantic content explicit. As a result, symbolic models based on SAA notation often resort to a patchwork of external fixes such as look-up tables, inference rules and arbitrary, hand-coded similarity matrices (e.g., Salvucci & Anderson, 2001; high-rank tensors, e.g., Halford et al., 1998, are a notable exception, but see Doumas & Hummel, in press, and Holyoak and Hummel, 2000, for detailed treatments of the limitations of this approach).

Importantly, these fixes are external to the relational representations themselves: They are not instantiated in the notation of the representations that encode relations, but rather only at the level of the system that *processes* these representations. This separation is fine for accounts of cognition at Marr's (1982) level of computational theory, but for accounts at the level of representation and algorithm such fixes entail specifying multiple sets of representations (at minimum one for the notational system capturing the relations and a second for the system that captures the relations between different relations and/or semantics of those relations), and an interface for moving between them.

Moreover, relations in SAA are represented as indivisible wholes, leaving the roles implicit as semantically empty place-holders or arcs in a graph. For example, the *love* (x, y) relation is represented by a single symbol (in propositional notation), a single vector (in high-rank tensor products), or a single node (in a labeled graph)[2], and its roles, *lover* (x) and

[2] Not all models that employ labeled-graphs are based on SAA (e.g., Larkey & Love, 2003), although most are.

beloved (y), are not represented explicitly at all. As such, even if a relational symbol (as a whole) is assumed to have semantic content, SAA notation itself does not specify how this content is differentially applicable to its arguments. It is this limitation that gives rise to the n-ary restriction: Even if the symbol *taller-than* is assumed to have meaning, the expression *taller-than* (x, y) does not explicitly specify how the role filled by x is semantically similar to *tall* (z), so it provides no basis for mapping *taller-than* (x, y) onto *tall* (z).

Recall the semantic relations between the roles of *murder* (x, y), *manslaughter* (x, y), and *attempted-murder* (x, y). SAA notation can, at best, treat the entire relations as simply "similar" or "different". High-rank tensor systems, for instance, represent entire relations (but not their roles) as vectors, and compute the similarities between relations based on the similarities of their vectors. Models using propositional notation and labeled graphs can "recast" predicates into more abstract forms (e.g., coding both *murder* (x, y) and *manslaughter* (x, y) as *commit-violence-against* (x, y)). Neither of these approaches expresses the similarity relations between the individual roles of relations, however, because they fail to make individual relational roles (and thus their semantic content) explicit. As a result, they do not make clear how the roles of *manslaughter* (x, y) and *attempted-murder* (x, y) are related to the roles of *murder* (x, y), nor how they differ.

It is easy to imagine a look-up table or set of inference rules that would supply this information, but the number of rules required to specify the similarity relations between all relational roles would scale minimally with $(n^2 - n)/2$ (or $n^2 - n$ if equivalent bidirectional similarity is not assumed), where n is the number of relational roles in the system. Worse, these rules would be external to the system's SAA-based representational system, so an account of how the system operates at the level of representation and algorithm requires a description of the rule set, and an additional control structure to read the SAA and access the rules as necessary. All of this is obviated in a system that simply codes the semantic content of individual relational roles explicitly in the notation of the relational representations.

The convenience of "postulate them as you need them" inference rules makes it is tempting to assume that the lack of role-specific semantic information in SAA is merely a thorny inconvenience: Surely the problem of role-based semantics in SAA is solvable, and will be solved as soon as it becomes important enough for someone to give it attention. In the meantime, it is certainly no reason to abandon SAA as a model of mental representation, especially if the only alternatives are non-symbolic representational schemes (to anticipate, they are not).

But it turns out that the problem is more than just a thorny inconvenience. As elaborated in the next section, role-specific semantic information cannot be made internal to (i.e., explicit in) the notation of SAA. Instead, the knowledge of the semantics of individual roles can only be specified at the level of the whole system, instantiated in external routines or representational structures. As a result, systems employing SAA (i.e., traditional symbolic models) are at best incomplete as algorithmic accounts of human cognition.

Representing Relational Roles in SAA

In a symbol system, representing something explicitly means representing it as a structure (e.g., a proposition). Thus, to represent relational roles and their semantic content explicitly in SAA, it is necessary to represent them structurally (i.e., to *predicate* them). For example, to represent the *murder* (x, y) relation in terms of its *killer* (x) and *victim* (y) roles, SAA must represent *killer* (x) and *victim* (y) as propositions (or tensors, or nodes), and simultaneously represent the fact that together they compose the *murder* (x, y) relation.

A relation specifies that its arguments are engaged in certain specific functions or states. For instance, the statement *drive* (Brutus, the Honda) specifies that Brutus is playing the role of the individual driving, and that the Honda is playing the role of the thing being driven. This information is entailed by the relational statement itself. As a separate matter, the relation may also imply or suggest other relations (e.g., that Brutus knows how to drive, that the Honda is in running condition, etc.), but it directly entails only the *driver* and *driven-object* roles, and the binding of Brutus and the Honda to these roles.

A relation does more than simply imply its roles (just as the roles do more than simply imply their parent relation), it *consists of* them: Collectively, the roles, along with their linkage to one another, *are* the relation. Imagine a relation R (p, q), with roles $R_1(p)$ and $R_2(q)$, that implies relation S (p, q), with roles $S_1(p)$ and $S_2(q)$. Although R (p, q) entails or implies a total of four roles (i.e., those of R and those of S), only two (R_1 and R_2) compose R itself. Therefore, representing relational roles explicitly requires explicitly specifying which relations consist of which roles.

As summarized above, this information seems to come gratis (i.e., without computational cost) as an integral component of human relational representations. An adequate model of human relational representations should, therefore, capture this information in its relational representations, rather than relegating it to the processes that operate on these representations or to external representational systems. This distinction is subtle, but important. It is not enough that the system as a whole capture this information; instead the same representations that encode the relations should capture it. That is, the information should be captured at the level of the notation, or language, of the system rather than in an external system of inference rules, look-up tables, etc.

However, it is not possible to capture relations, their roles and the relation between them in the notation of SAA. In SAA representing a relation explicitly requires instantiating a proposition. If representing a relation implies representing its roles, and representing a relation's roles requires representing the relation between the relation and its roles, then representing any given relation implies a second proposition representing the relation between the first relation and its roles. This second proposition contains a new relation, though, which implies roles of its own that must be related back to it in a third proposition, which also contains a new relation with roles that must be related back to it, and so on.

For example, imagine the propositions *love* (John, Mary) and *love* (Bill, Sally) and their role bindings, *lover* (John), *beloved* (Mary), *lover* (Bill) and *beloved* (Sally). The representations must specify which role bindings go together to form which relations (e.g., that *lover* (John) goes with *beloved* (Mary) – as opposed to *beloved* (Sally) or *lover* (Bill) – to form the complete relation *love* (John, Mary)). Specifying this composition relation in SAA notation requires a proposition of the form *consists-of* (*love* (John, Mary), *lover* (John), *beloved* (Mary))[3]. The predicate *consists-of*, however, is itself a relation, and so specifies a set of role bindings of its own, which must be explicitly represented and linked back to the original *consists-of* relation with a second *consists-of* relation. This second *consists-of* is also a relation and so specifies a set of role bindings of its own that must be linked back to it via a third *consists-of* relation, and so on.[4] The consequence is an infinite regress that can render the resulting representational system ill-typed (Manzano, 1996).

In other words, it is not possible to represent the relation between relational roles and complete relations explicitly in the notation of SAA. For the same reason, it is not possible to specify the semantic content of relational roles explicitly in SAA. As noted previously, this information can be specified in a complete *system* based on SAA (after all, many such systems are Turing complete), but it is not possible to capture this information in the SAA notation itself. Instead, the knowledge is necessarily contained in the processes that operate on the SAA representations, or in external representational systems that must be interfaced with the SAA notation. This property renders SAA notation fundamentally inadequate as a model of human mental representations. It does not imply that representing the semantic content of relational roles explicitly is impossible in an SAA based system; it simply indicates that such systems cannot capture this content in the same way that people seem to—i.e., as an integral part of the relational representation itself.

It is important to emphasize that the problem of infinite regress is by no means an argument against the general utility of SAA. On the contrary, propositional notation, for example, is an extremely useful tool for the purpose for which it was created, namely theorem proving. For this purpose, semantic emptiness is a virtue. It just happens that the design requirements for a representational system for theorem proving differ markedly from the design requirements for a representational system for supporting perception and cognition in living, behaving systems: While the former requires pristine semantic emptiness, the latter must deal with the semantically-rich, sometimes ugly realities of the world as it is. Given this, it is not surprising that a representational system designed for theorem proving should prove inadequate as a model of human mental representation.

Possible Objections and Responses

We are arguing that SAA is ill-suited for modeling human mental representations, including relational representations, for which it appears at first to be ideally suited. Proponents of traditional symbolic models of cognition are likely to object strenuously to our arguments. We shall try to anticipate some of these objections and respond to them.

One potential objection is that the infinite regress is irrelevant because it is a trivial matter to simply terminate the regress at any arbitrary point and declare it done. The problem with this objection is that the system does not finish the process it began. Once the progress halts, the relation between roles and relations remains unspecified.

Another potential objection is that although role-specific information cannot be specified in the SAA notation itself, it might be adequate to have the information specified at the level of the system as a whole. The problem with this objection is that it is an appeal to the status quo: Using lookup tables, etc., is precisely what current SAA-based models are forced to do. As we have argued, the data on the role of semantics in human cognition suggest that this approach is not adequate. Among other problems, it is too deliberate: The data on the role of semantics in memory retrieval, analogical mapping, etc., suggest that people use role-based semantic information automatically and without computational cost. Specifying information at multiple levels of representations and postulating routines for moving between the two is computationally costly. Although it is impossible, given the current state of our knowledge, to rule such an account out definitively, it certainly seems more plausible to assume that role-based information is simply a part of relational representations than to assume that the mind is equipped with a complex system of look-up tables. Indeed, the use of lookup tables is perhaps even more awkward than it appears at first blush. Without an explicit representation of role-specific semantics in its representations, an SAA system must use an external meta-system to interpret its SAA representations, decide when the look-up table (another representational system external to SAA) should be accessed, retrieve the relevant role information, translate that information into SAA, and insert it into the original SAA format. Importantly, the external control structure could not, itself, utilize SAA: if it

[3] Naming the relation that specifies the relation of a relation to its roles *consists-of* is, of course, completely arbitrary.

[4] One might argue that this problem could be solved by using binary rather than unary representations of roles. For example, representing *lover* (John) as *lover* (John, *loves* (John, Mary)), thus representing the role, its filler, and the relation to which it belongs. The problem with this approach is that it simply transfers the same problem to a new representation. In SAA binary predicates dictate relations. As a result, the system would still have to specify the roles of this binary predicate and how they relate back to it.

did, then it too would require an external meta-system to keep it's own relation and role information straight. This solution is thus both inelegant and impractical, and in the limit results in a regress, not of nested "consists-of" relations, but of external control structures.

A third objection to our argument concerns the *n*-ary restriction. This problem might be solved simply by postulating that all *n*-place predicates (where *n* is free to vary) be replaced by *m*-place predicates (where *m* is a constant, say, 3). For example, *tall* (*x*) would be recast as *tall* (*x*, -, -), and *taller-than* (*x*, *y*) as *taller-than* (*x*, *y*, -), where "-" denotes a permanently empty "dummy" slot. In this way, *tall* (*x*, -, -) could be mapped to *taller-than* (*x*, *y*, -) by virtue of their both taking three "arguments". One problem with this proposal is that it assumes some procedure for deciding how to recast the predicates. For example, should *short* (*y*) be recast as *shorter-than* (*y*, -, -) (the intuitive recasting) or as *taller-than* (-, *y*, -)? Note that this question must be answered *prior* to discovering the very mapping(s) the recasting is supposed to permit. A second problem with this proposal is that it still leaves the problem of mapping non-identical predicates: How should the system map the arguments of *taller-than* (*x*, *y*, -) to those of *shorter-than* (*x*, *y*, -) without knowing what the roles of each relation "mean"?

Semantically Rich Representations of Relational Roles

We have argued that SAA is at best incomplete as a model of human mental representation. However, this limitation does not by any means imply that we must abandon symbolic accounts of mental representation. Rather, what is needed is a representational system that makes relational roles, their semantics, their bindings to their fillers and their composition into complete relations all explicit. This approach is commonly known as *role-filler binding* (e.g., Halford, et al. 1998).

In a role-filler binding system roles are represented explicitly and bound to their fillers. Relations are composed of linked role-filler bindings. One example is the representational system employed by Hummel and Holyoak's (1997, 2003a) LISA model. LISA uses a hierarchy of distributed and localist codes to represent relational structures. At the bottom, "semantic" units represent objects and roles in a distributed fashion (e.g., for the relation *love* (John, Mary) "John" might be represented as *human*, *adult*, *male*, etc., and Mary by *human*, *adult*, *female*, etc.; similarly, the *lover* and *beloved* roles of the *love* relation would be represented by units capturing their semantic content). At the next level, these distributed representations are connected to localist units representing individual objects and relational roles. Above the object and role units, localist role-binding units (SPs) link object and role units into specific role-filler bindings. At the top of the hierarchy, localist P units link SPs into entire propositions.

In this representation, the long-term binding of roles to their fillers is captured by the conjunctive SP units, thus violating the role-filler independence required for symbolic representation. However, when a proposition becomes active, its role-filler bindings are also represented dynamically by synchrony of firing: SP units in the same proposition fire out of synchrony with one another, causing object and predicate units, along with their corresponding semantics, to fire in synchrony with each other if they are bound together, and out of synchrony with other bound roles and objects. On the semantic units, the result is a collection of mutually desynchronized patterns of activation, one for each role-filler binding. Thus, when a proposition is active, role-filler independence is maintained on its object, predicate and semantic units, with the role-filler bindings carried by the synchrony relations among these units.

This is only one way to represent relational knowledge in a role-filler binding scheme (see also Halford et al, 1998, Smolensky, 1990). As demonstrated by Hummel, Holyoak and their colleagues (see Hummel & Holyoak, 2003b for a review), however, it is a very useful one. LISA has been shown to simulate aspects of memory retrieval, analogical mapping, analogical inference, schema induction, and the relations between them. It also provides a natural account of the capacity limits of human working memory, the effects of brain damage and normal aging on reasoning, the relation between effortless ("reflexive") and more effortful ("reflective") reasoning, and aspects of the perceptual-cognitive interface. It inherits these abilities from its ability to capture both the relational structure and the semantic richness of human mental representations.

Conclusion

Few would claim that people literally think in the predicate calculus. However, many researchers have argued that SAA-based representations serve *at least* as a plausible shorthand for human mental representations. We have argued that SAA notation is *at most* a shorthand for human relational representations—a shorthand that must necessarily leave the messy business of the semantics of relational roles to fundamentally external representations and processes. Inasmuch as the semantics of roles are an important internal component of human mental representation, as we, and others, have argued they are, this fact leaves an important facet of human mental representation necessarily beyond the reach of SAA-based models. The solution to this problem is not to abandon symbolic representations altogether, as proposed by some, but rather to replace SAA with a role-filler binding approach to the representation of relational knowledge in models of cognition. Doing so provides a natural basis for capturing both the symbolic structure of human mental representations and their semantic richness.

Acknowledgements

The authors would like to thank Keith Holyoak, Keith Stenning, Ed Stabler, Tom Wickens, and Charles Travis for helpful discussion and comments. This work was supported by a grant from the UCLA Academic Senate.

References

Anderson, J. R. & Lebiere, C. (1998). *The atomic components of thought*. Mahwah, NJ: LEA.

Bassok, M., Wu, L., & Olseth, K. L. (1995). Judging a book by its cover: Interpretative effects of content on problem-solving transfer. *Memory & Cognition*, 23, 354-367.

Doumas, L. A. A., & Hummel, J. E. (in press). Modeling human mental representations: What works, what doesn't, and why. In K. J. Holyoak & R. Morrison (Eds.), *The Cambridge Handbook of Thinking and Reasoning*. Cambridge: Cambridge University Press.

Eliasmith, C. & Thagard, P. (2001). Integrating structure and meaning: A distributed model of analogical mapping. *Cognitive Science*, 25, 245-286.

Elman, J. L., Bates, E., Johnson, M. H., Karmaloff-Smith, A., Parisi, D., & Plunkett, K. (1996). *Rethinking Inateness: A Connectionist Perspective on Development*. Cambridge, MA: MIT Press.

Faulkenhainer, B., Forbus, K. D., & Gentner, D. (1989). The structure mapping engine: Algorithm and examples. *Artificial Intelligence*, 41, 1-63.

Forbus, K. D., Gentner, D., & Law, K. (1995). MAC/FAC: A model of similarity-based retrieval. *Cognitive Science*, *19*, 141-205.

Gentner, D., Ratterman, M. J., & Forbus, K. D. (1993). The roles of similarity in transfer: Separating retrievability from inferential soundness. *Cognitive Psychology*, 25, 524-575.

Halford, G. S., Wilson, W. H., & Phillips, S. (1998). Processing capacity defined by relational complexity: Implications for comparative, developmental, and cognitive psychology. *Brain and Behavioral Sciences*, *21*, 803-864.

Holyoak, K. J., & Hummel, J. E. (2000). The proper treatment of symbols in a connectionist architecture. In E. Dietrich and A. Markman (Eds.). *Cognitive Dynamics: Conceptual Change in Humans and Machines* (pp. 229 - 264). Hillsdale, NJ: Erlbaum.

Holyoak, K. J., & Thagard, P. (1989). Analogical mapping by constraint satisfaction. *Cognitive Science*, *13*, 295-355.

Hummel, J. E., & Holyoak, K. J. (1997). Distributed representations of structure: A theory of analogical access and mapping. *Psychological Review*, 104, 427-466.

Hummel, J. E., & Holyoak, K. J. (2003a). A symbolic-connectionist theory of relational inference and generalization. *Psychological Review*, 110, 220-263.

Hummel, J. E., & Holyoak, K. J. (2003b). Relational reasoning in a neurally-plausible cognitive architecture: An overview of the LISA project. *Cognitive Studies: Bulletin of the Japanese Cognitive Science Society, 10*, 58-75.

Keane, M. T., Ledgeway, T., & Duff, S. (1994). Constraints on analogical mapping: A comparison of three models. *Cognitive Science, 18*, 387-438.

Krawczyk, D. C., Holyoak, K. J., & Hummel, J. E. (in press). Structural constraints and object similarity in analogical mapping and inference. *Thinking and Reasoning.*

Kubose, T. T., Holyoak, K. J., & Hummel, J. E. (2002). The role of textual coherence in incremental analogical mapping. *Journal of Memory and Language, 47*, 407-435.

Larkey, L. & Love, B. (2003). CAB: Connectionist Analogy Builder. *Cognitive Science* , 27, 781-794.

Landauer, T.K. and Dumais, S. T. (1997) A solution to Plato's problem: TheLatent Semantic Analysis theory of acquisition, induction andrepresentation of knowledge. *Psychological Review*, 104, 211-240.

Manzano, M. (1996). *Extensions of first order logic*. Cambridge: Cambridge University Press.

Marcus, G. F. (1998). Rethinking eliminative connectionism. *Cognitive Psychology*, 37(3), 243-282.

Marr, D. (1982). *Vision*. Freeman: San Francisco.

Newell, A. (1990). *Unified theories of cognition*. Cambridge, MA: Harvard University Press.

Ramscar, M. & Yarlett, D. (2003) Semantic Grounding in Models of Analogy: An Environmental Approach. *Cognitive Science, 27*, 41-71.

Ross, B. (1987). This is like that: The use of earlier problems and the separation of similarity effects. *JEP: Learning, Memory, and Cognition*, *13*, 629-639.

Ross, B. (1989). Distinguishing types of superficial similarities: Different effects on the access and use of earlier problems. *JEP: Learning, Memory, and Cognition*, *15*, 456-468.

Rumelhart, D. E., McClelland, J. L., & the PDP Research Group (1986). *Parallel distributed processing: Explorations in the microstructure of cognition*. (Vol. 1). Cambridge, MA: MIT Press.

Salvucci, D. D., & Anderson, J. R. (2001). Integrating analogical mapping and general problem solving: The path mapping theory. *Cognitive Science*, 25, 67-110.

Smolensky, P. (1990). Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence*, *46*, 159-216.